Research Article

pKAS: A Secure Password-Based Key Agreement Scheme for the Edge Cloud

Ping Liu,1 Syed Hamad Shirazi,2 Wei Liu,1 and Yong Xie1

1Department of Computer Technology and Application, Qinghai University, Xining, China
2Department of Information Technology, Hazara University, Baffa, Pakistan

Correspondence should be addressed to Yong Xie; mark.y.xie@qq.com

Received 5 September 2021; Accepted 5 October 2021; Published 18 October 2021

Copyright © 2021 Ping Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Forthesimplicityandfeasibility,password-basedauthenticationandkeyagreementschemehasgraduallybecomeapopularway
toprotectnetworksecurity. In order to achieve mutual authentication between users and edge cloud servers during data
collection, password-based key agreement scheme has attracted much attention from researchers and users. However, security
and simplicity are a contradiction, which is one of the biggest difficulties in designing a password-based key agreement scheme.
Aimedtoprovidesecureandefficientkeyagreementschemesfordatacollectinginedgecloud,weproposeanefficientandsecure
key agreement in this paper. Our proposed scheme is proved by rigorous security proof, and the proposed scheme can be
protected from various attacks. By comparing with other similar password-based key agreement schemes, our proposed scheme
has lower computational and communication costs and has higher security.

1. Introduction

With the dawn of the Internet of everything, Internet of things (IoT) has become to obtain the leading strategic
position in research and development in the world. Even though various countries in the world pay attention to the
development of the IoT, the influx of diverse traffic and the need of diversified application scenario has not only put
forward new challenge for the centralized cloud computing architecture nowadays but also drove the emergence of the
cloud computing paradigm [1, 2].

In the era of Internet of Things, mobile devices are no longer simple mobile phones, tablets, etc., but include more
abundant augmented/virtual reality devices, intelligent medical device, and moving vehicle. The application scen-
ario also transfers from voice/video communication and other services to virtual space experience, intelligent
manufacturing, and the Internet of vehicles [3, 4]. In cloud-based services, data transmission speed will be affected by
network traffic, and heavy traffic will lead to long transmission time, thus increasing power consumption cost.

Therefore, the adoption of mobile edge computing (MEC) can meet the needs of IoT devices.

As shown in Figure 1, the collection and processing of data is a very important part of the Internet of Things. However, all collected data will be transmitted to the cloud server and then rely on the server’s computing power for
dataprocessingandanalysis. This will cause the server to be heavily loaded and prone to failure or downtime. At the
same time, the increase in the amount of data will also increase the cost of the storage server. In addition, because
the network is limited by the network bandwidth and speed, the network bandwidth is put under pressure when a large
amount of monitoring data is transmitted, and the data may have large transmission delays and packet loss during
transmission. Edge computing data provides format conversion, caching, processing, analysis, and transmission
services, and the load of cloud servers improves the efficiency of data processing. The edge cloud includes IoT
gateways and collectors. These devices together form an edge node network and provide lightweight computing power for
the edge layer of the system.
In the MEC-based Internet of Things, massive amounts of data are generated by a large number of sensors and various heterogeneous devices, and all storage devices are provided by different third-party vendors. Due to the distributed nature of MEC, data are stored in different network edges, which will increase the risk of data being attacked. For example, unauthorized users or opponents may modify or abuse the data uploaded in the storage, which will lead to data leakage and other problems. In order to solve these problems, this paper proposes identity verification based on password-based key agreement. This scheme can ensure both sides' identity authentication and data security.

In order to protect the data in the edge cloud from being tampered with, the administrator of the edge cloud server needs to authenticate with it when operating the server, so the server can determine whether the administrator has been faked. To improve the security and verifiability of messages, Zheng [5] proposed a signcryption scheme, which can simultaneously sign and encrypt.

The key agreement protocol is the most commonly used method for two or more parties to communicate. Features of the protocol ensure that the data to be communicated are confidential, secure, and complete [6–10]. The protocol is to establish a session key jointly by two or more entities. The result of key agreement will be affected by any participant, and no trusted third party is required in the process. The session key is obtained by calculating the parameters generated by the participants. In order to enable both parties to authenticate each other, an authentication key agreement is proposed, and the protocol established a session key [11–13].

In 2005, the Diffie–Hellman key exchange in the encryption assumption protocol system is a secure and scalable authentication key exchange agreement, which performs key control and management during transmission [14–16]. In 2009, the elliptic curve cryptosystem (ECC) authentication scheme based on no pairing and few certificates was presented. The scheme was based on mobile devices communication and ID authentication with key agreement protocol. Furthermore, the proposed scheme is also to overcome more attacks [13, 17–20]. Many scholars believed that large prime numbers is difficult for hardware implementation of the elliptic curve cryptosystem, while the binary field was known as suitable [21, 22] in 2010–2012. In order to ensure the confidentiality and integrity of the sent and received messages, the authentication key agreement protocol must include a strong encryption algorithm. The key agreement protocol based on elliptic curve cryptography provides an important development for confidentiality, integrity, and user anonymity.

There are two types of key agreement protocols according to different authentication methods: password-based key agreement protocols and public-key-based key agreement protocols. The password-based authentication key agreement protocol was first proposed by Bellovin and Merritt [23]. In this protocol, both parties share a password in advance, which is used to authenticate each other's identity during communication and negotiate a short-term
session key. Public key-based key agreement can negotiate a session key through signature or public key verification. In this paper, password-based key agreement protocol is studied [15].

1.1. Motivations and Contributions. The proposed pKAS can ensure the security of the message and the authentication of the user identity when two parties communicate. We list our contributions as follows:

First, we put forward a secure password-based key agreement pKAS based on ECC for mutual authentication between the user and edge server. The proposed pKAS only needs to deliver the message twice, which greatly saves communication bandwidth. And, in this scheme, we use signcryption, signature verification, and hash operation etc., to ensure the confidentiality and integrity of the message, as well as the anonymity of the identity.

Second, we conduct strict security analysis on the proposed pKAS and compare it with other related schemes. The results show that the presented pKAS can resist various attacks.

Third, by comparing communication and calculation costs, the proposed pKAS has lower cost and is more secure than recent similar schemes.

1.2. Organization of the Paper. The structure of the paper is as follows. Sections 2 and 3 present the related works and the preliminaries. The system model and security requirements of the scheme proposed in this paper are shown in Section 4. Section 5 presents the proposed password-based key agreement scheme. Section 6 presents the performance and security analysis. Section 7 describes conclusion, future work, conflicts of interest, and data availability respectively.

2. Related Works

With the development of Internet technology, security in communications has become more and more significant. Therefore, how to identify remote users has become one of the most significant issues in the public network. In order to figure out the problem, many schemes have been presented. Lamport [24] first proposed the password-based scheme to ensure remote parties authentication scheme. Subsequently, many password-based key agreement schemes were proposed in [25–29].

In 2009, Xu et al. [25] presented an improved remote user authentication and key agreement scheme based on passwords and smart cards, and they certificated that their scheme is secure. Sood et al. [26] found that Xu et al.’s scheme is ineffective against password guessing attacks and impersonation attacks. Subsequently, Sood et al. put forward an improved authentication scheme. However, in 2012, Chen et al. [27] analyzed and pointed out that the scheme of Sood et al. only provided a single-party authentication function, and the legitimacy of the remote server was not authenticated. As a consequence, an improved key agreement scheme with stronger security was presented by Chen et al.

3. Preliminaries

3.1. One-Way Hash Function. Let message $m$ be a message that requires a hash value. The length of $m$ is a variable, while $h$ is the fixed length. Given $m$, it is easy to obtain $h$. However, given $h$, it is infeasible to obtain $m$. 

$h \leftarrow \text{hash}(m)$
3.2. Elliptic Curve Cryptosystem (ECC). In 1985, the elliptic curve was used for data encryption by Miller firstly. Later, Koblitz based on the elliptic curve discrete logarithm problem (ECDLP) built a new encryption system, which is called the elliptic curve cryptosystem (ECC). ECC has lower computational overhead than other public key cryptographies such as RSA. Since then, ECC has been widely used in cryptographic protocols and security schemes. The following describes the basic knowledge of ECC and computational difficulties in ECC.

Elliptic curve cryptography is a public key cryptography method based on elliptic curve mathematics. The commonly used expression of elliptic curve in finite field $F_p$ is: $y^2 = x^3 + ax + b (\mod p)$ and $(4a^3 + 27b^2) (\mod p) \neq 0$, all coefficients are elements in a finite field $F_p$ (where $p$ is a large prime number). Let $E_p(a, b)$ denote the point set $\{(x, y)|0 \leq x < p, 0 \leq y < p, \text{ and } x, y \text{ are both integers}\}$ on the elliptic curve defined by the equation and the infinity point $O$.

The addition on $E_p(a, b)$ is defined as follows:
For any point in $E_p(a, b)$, $P = O$.
Let $Q, R$ be the two points in $E_p(a, b)$, $Q + R$ is defined as follows: draw a straight line passing through $Q, R$ and the elliptic curve to intersect point $P$, then $Q + R = -P$.

Let $Q$ be a point in $E_p(a, b)$, and the multiples of $Q$ are defined as follows: draw a tangent to the elliptic curve at point $Q$, and set the tangent to intersect the elliptic curve at point $S$; then, $2 \cdot Q = Q + Q = -S$. Similarly, $n \cdot Q = Q + Q + \cdots + Q (n$ times), where $n \in Z_p, n > 0$.

3.3. Complexity Assumptions. The security foundation of ECC is an elliptic curve discrete logarithm problem (ECDLP), which can be defined as follows.

ECDLP: assume two random points $P_1$ and $P_2$ in $(E/F_p)$, $P_2 = kP_1$, where $k \in Z_p$. It is easy to compute $P_2$ if knows $k$ and $P_1$, while it is infeasible to compute $k$ if knows $P_1$ and $P_2$.

4. System Model and Security Model

4.1. System Model. On analysis of the requirements of communication between the user and edge server, there are two types of roles related in our system, such as users communicating with server, a trust authority (TA) can be regarded as a completely trusted administrator and cannot be compromised by any adversary. With a view to user authentication and key agreement, a user (Assumed be $U_i$) must be registered in the TA, and then he/she can perform mutual authentication and key agreement with edge cloud server other users (such as $U_j$) only using the password and smart card.

The network model of our system can be illustrated in Figure 2. Before the users communicate with the edge server, the users must register with the TA through a secure channel and store the corresponding registration information on her/his smart cards. After successful registration, users can perform mutual authentication and key negotiation through edge server and implement operations such as secure data management on the edge cloud.

4.2. Security Requirements. Before analyzing security requirements, let us assume adversary’s capabilities based on the application. An adversary $A$ generally contains the following capabilities:

(i) The open channel can be controlled by $A$, that is to say, the messages through the open channel $A$ can be deleted, intercepted, modified, and resent

(ii) $A$ can traverse the password space in polynomial time, that is, if it has known any other secret information, $A$ can guess the password by brute force attack

(iii) $A$ can obtain the user’s password through a malicious terminal and can also extract data that are stored in smart card

On the capacities of the adversary $A$, the security requirements of password-based key agreement scheme should include forward secrecy and must be resistant to know attacks, such as offline password guessing attack, replay attack, user impersonation attack, server spoofing attack, and parallel attack. Furthermore, the scheme must be mutual authentication and anonymity.

5. The Proposed Scheme (pKAS)

In this section, a key agreement scheme based on password (called pKAS for short) by using ECC was proposed. There are no bilinear paring operations in pKAS. Overall, pKAS has four phases: system initial phase, registration phase, login and key agreement phase, and offline password change phase. For simplicity, we list the symbols used in this paper and their corresponding meanings in Table 1.

Next, the following sections present the four phases of the proposed scheme.

5.1. System Initialization Phase. Trust authority (TA) is responsible for the system initialization phase. In this phase,
TA selects a big prime \( p \); then, in finite field, \( F_p \) constructs a nonsingular elliptic curve \( E_p(a,b) \) and chooses base points \( P \) on \( E_p(a,b) \) and generates a finite cycle additive group \( G \) of order \( q \) with \( p \).

### 5.2. Registration Phase

Users, edge cloud sever, and TA complete the registration phase together. Assume the current user \( U_i \)’s identity be \( ID_i \), the registration is completed as follows:

Step R1: \( U_i \) sets his password \( PW_i \), then chooses a random number \( x_i, a_{i0} \in Z_q^* \), and computes \( X_i = x_iP, a_{i1} = h(PW_i \| a_{i0}) \). At last, \( U_i \) sends \( ID_i \), \( X_i \) to TA in a channel that an adversary cannot eavesdrop on.

Step R2: when TA receives \( ID_i \), \( X_i \), it will store \( \{ID_i, X_i\} \) in the server.

### 5.3. Login and Key Agreement Phase

We assume there are two users, user \( U_i \) and edge cloud sever \( U_j \) in this phase. They log in by using their ID and password, then authenticate, and consult with session key each other.

Step A1: \( U_i \) inputs his/her ID \( ID_i \) and \( PW_i \), then smart card computes \( a_i = h(ID_i) \| h(PW_i \| a_{i0}) \mod \kappa \), and checks whether \( a_i = a_i \) holds or not. If it does not, the session is terminated.

Step A2: \( U_i \) randomly chooses \( c_i \in Z_q^* \), computes \( C_i = c_iP \),

\[
PID_i = ID_i \| h(c_i \| X_i \| t_i),
\]

\( f_i = h(ID_i \| PID_i \| C_i \| t_i) \), where \( t_i \) is current timestamp, \( \sigma_i = c_i + x_i f_i \mod q \). At last, \( U_i \) sends \( M_1 = \{c_i, PID_i, t_i, \sigma_i\} \) to \( U_j \).

Step A3: after receiving \( \{C_i, PID_i, t_i, \delta_i\} \), \( U_j \) checks whether \( t_j - t_i < \Delta t \), if not, \( U_j \) terminates the session, else \( U_j \) computes \( C_j = \sigma_iP - f_iX_j, \)

\[
sk_{j} = h(ID_i \| t_i \| C_i \| C_j \| t_j), t_i \| f_i \| ID_i \| C_i \| C_j \| t_j)
\]

\( \delta_j = C_j + x_j f_j \mod q \). At last, \( U_j \) sends \( M_2 = \{C_j, f_j, \delta_j, t_j\} \) to \( U_i \).

Step A4: after receiving \( \{C_j, f_j, \delta_j, t_j\} \) from \( U_j \), \( U_i \) checks whether current timestamp \( t_i \) meets \( t_j - t_i < \Delta t \) or not, if not, \( U_i \) terminates the session, else \( U_i \) computes

\[
\begin{align*}
C_i' &= \sigma_iP - f_jX_i, \\
\delta_i &= C_i + c_jX_i, \\
\delta_j &= C_j + x_i f_j \mod q, \\
\end{align*}
\]

At last, \( U_i \) and \( U_j \) have agreed an identical session key \( sk = sk_j \). Figure 3 presents the flowchart of login and key agreement phase.

### 5.4. Offline Password Change Phase

In order to obtain a better user experience, while meeting the high requirements of security and efficiency, the user can complete this phase locally in the proposed scheme as follows:

Step C1: in order to verify the user’s identity, the user must enter \( ID_i \), \( PW_i \) in the smart card.

Step C2: the smart card computes \( a_i' = h(h(ID_i) \| \delta_i \| PW_i \| a_{i0}) \mod \kappa \) and checks if \( a_i' \) and \( a_i \) are equal. If not, the system will terminate the session. Else, it means the correctness of \( ID \) and \( PW_i \), \( \kappa = 2^\kappa \), and it can go to the next step.

Step C3: user \( U_i \) inputs new password \( PW_i^{\text{new}} \) and computes \( a_i = h(h(ID_i) \| \delta_i \| PW_i^{\text{new}} \| a_{i0}) \mod \kappa \).

### 6. Security and Performance Analysis

Security analysis and proof of our scheme is presented in this section. As well as the proposed pKAS is proven to be able to resist all kinds of attacks. Besides, we analyze and compare the communication calculation and bandwidth consumption of similar schemes.

#### 6.1. Security Analysis

In this section, the details of security analysis are described as following.

**Proposition 1.** The proposed pKAS scheme can be secure against offline password guessing attack.

**Proof.** Assume an adversary \( \mathcal{A} \) has got \( U_i \)’s smart card and obtained the data stored in the card, he/she can launch password guessing attack by the following steps:

Step D1: \( \mathcal{A} \) guesses \( PW_i \) from password dictionary space and \( ID_i \) from identity dictionary space

Step D2: \( \mathcal{A} \) retrieves \( a_{i0} \) and \( a_i \) and computes \( a_i' = h(h(ID_i) \| \delta_i \| PW_i^{\text{new}} \| a_{i0}) \mod \kappa \)

Step D3: \( \mathcal{A} \) checks whether \( a_i' = a_i \) holds or not

Step D4: \( \mathcal{A} \) repeats the step D1 to D3 until \( a_i' = a_i \) holds

That is, \( \mathcal{A} \) can guess correct ID, and \( PW_i \). However, \( \mathcal{A} \) is still not sure they are the same identity and password. Then, \( \mathcal{A} \) has to execute online guessing attack to test the correctness both. However, we use Hoeney_list to prevent online...
guessing attack. As a result, the proposed pKAS can be secure against offline password guessing attack.

**Proposition 2.** The proposed pKAS scheme can be secure against online password guessing attack.

**Proof.** In order to eliminate the threat of online password guessing attack, Hoeny list is adopted in the proposed scheme. As analysis of Proposition 1, the proposed pASK can use Hoeny list to prevent online guessing attack. Therefore, the proposed pKAS scheme can be secure against online password guessing attack.

**Proposition 3.** The proposed pKAS scheme can provide anonymous interactions among the users $U_i$ and edge cloud sever $U_j$, and no adversary $\mathcal{A}$ can obtain both identity information during login and key agreement phase.

**Proof.** In the login and key agreement phase of pKAS, user $U_i$'s real identity $ID_i$ is hidden in message $PID_i = ID_i \oplus h(c_i, X_j, t_i)$. If an adversary $\mathcal{A}$ can reveal the $ID_i$ from the messages, he/she should solve the ECDLP problem because PID, include ECDLP in their construction. Therefore, the proposed pKAS can provide anonymous interactions during user login and key agreement.

**Proposition 4.** The proposed pKAS scheme can provide forward secrecy during the session key agreement.

**Proof.** Assume an adversary $\mathcal{A}$ has obtained the smart card and user’s password and identity. However, $\mathcal{A}$ cannot retrieve the previously existing session key without knowing $c_i$ because $\mathcal{A}$ should solve the ECDLP problem. Hence, the proposed pKAS scheme can give strong forward secrecy.

**Proposition 5.** The proposed pKAS scheme can be secure against forgery attack.

**Proof.** In the proposed scheme, $U_j$ can check that message $M_1$ has been forgery by computing $ID^* = PID^* \oplus h(x_j, C_j, t_j)$, $f_i' = h(ID^* || PID^* || C_j || t_j)$, and checking $C_i = \sigma_i P - f_i' X_j$ holds or not. $U_i$ authenticates $U_j$ by computing $C_i' = \sigma_i P - f_i' X_j$, $sk_{ij} = h(ID_i || ID_j || t_i || t_j || C_i || C_j)$ and checking $f_j = h(ID_i || C_i || sk_{ij} || t_i || t_j || ID_j)$ holds or not. When $\mathcal{A}$ modifies the message during the conversation, the tampered message cannot be verified. As a consequence, the proposed pKAS scheme can be secure against forgery attack.

**Proposition 6.** The proposed pKAS scheme can provide mutual authentication.

**Proof.** In the presented scheme, $U_j$ and $U_i$ verify message $M_1$ and $M_2$ by checking equation $C_i = \sigma_i P - f_i' X_j$, $f_i' = h(ID^* || PID^* || C_j || t_j)$, and checking $C_i' = \sigma_i P - f_i' X_j$, $sk_{ij} = h(ID_i || ID_j || t_i || t_j || C_i || C_j)$ and checking $f_j = h(ID_i || C_i || sk_{ij} || t_i || t_j || ID_j)$ hold or not, respectively. If it holds, the scheme achieves mutual authentication based on Proposition 5 that no adversary can successfully implement a forgery attack. Therefore, the presented pKAS scheme can give mutual authentication.

**Proposition 7.** The proposed pKAS can be secure against replay attack.

**Proof.** In the presented scheme, $U_j$ and $U_i$ verify message $M_1$ and $M_2$ by checking equation $C_i = \sigma_i P - f_i' X_j$, $f_i' = h(ID^* || PID^* || C_j || t_j)$, and checking $C_i' = \sigma_i P - f_i' X_j$, $sk_{ij} = h(ID_i || ID_j || t_i || t_j || C_i || C_j)$ and checking $f_j = h(ID_i || C_i || sk_{ij} || t_i || t_j || ID_j)$ hold or not, respectively. If it holds, the scheme achieves mutual authentication based on Proposition 5 that no adversary can successfully implement a forgery attack. Therefore, the presented pKAS scheme can give mutual authentication.
Proof. In the proposed pKAS scheme, we use timestamps and random numbers to prevent replay attack. Messages $M_1$ and $M_2$ include timestamps $t_i$ and $t_j$, respectively, which is a classic way to stop replay attacks. Random numbers are also used to prevent relay attack because users and server can check the validity of random number by verification algorithm each time and adversary $A$ still cannot construct valid session key. Hence, the presented pKAS can be secure against replay attack.

Proposition 8. The proposed pKAS can be secure against impersonation attack.

Proof. Let $A$ can get $U_i$’s smart card and know the data in the card by some way. However, $A$ has to possess PW, and ID into smart card to generate a legal message $M_1 = \{C_i, PID, a_i, t_i\}$. Without the two factors (PW, and ID), $A$ cannot compute a correct $a_i$ to pass the verification of smart card that $A$ cannot proceed to the next step to impersonate $U_i$ to communicate with other. Therefore, the proposed pKAS can security resist impersonation attack.

Proposition 9. The proposed pKAS can be secure against parallel attack.

Proof. Parallel attack usually occurs when an adversary $A$ constructs a new conversation to impersonate a legal user by reusing historical messages that he/she intercepted in a public channel. However, $A$ should know the parameters of messages or he/she cannot send a correct access request and gain a session key. However, $A$ cannot obtain the random number that is chosen by users. As a result, the proposed pKAS can be secure against parallel attack.

Proposition 10. The proposed pKAS can be secure against insider attack.

Proof. As shown in the user registration phase, user $U_i$ send \{ID, $X_i$\} to $U_j$, where $X_i = x_iP$. Without knowing $x_i$, the server cannot impersonate $U_i$. Therefore, the proposed pKAS can be secure against insider attack.

Proposition 11. The proposed pKAS scheme can achieve user untraceability.

Proof. In the proposed scheme, user $U_i$’s real identity ID, real identity ID, are hidden in message PID = ID $\oplus $ $h(c_iX_i, t_i)$. Only when an adversary $A$ can solve the ECDLDP problem, $A$ can reveal ID, from the messages that are included by ECDLDP in their construction. As a consequence, the proposed pKAS can achieve user untraceability.

Proposition 12. The proposed pKAS scheme can achieve key agreement.

Proof. $U_j$ computes his/her session key as $sk_{ij} = h(ID, [ID, t_j, C_j])$, in the step A3. $U_i$ computes his/her session key as $sk_{ij} = h(ID, [ID, t_i, C_i])$, in step A4. Because $c_jC_i = c_iC_j = c_{ij}P$, $U_i$ and $U_j$ can compute an identical session key $sk_{ij} = sk_{ij}$. Therefore, the proposed pKAS scheme can achieve key agreement.

Proposition 13. The proposed pKAS scheme can achieve offline password change.

Proof. As shown in introduction of the proposed scheme, offline password change phase is provided. Each user can achieve password change locally. If user inputs correct ID and PW, the correctness of ID and PW is $\kappa - 1/\kappa \approx 99.61/100, \kappa = 2^8$, i.e., user has a high probability of completing password local change. As a consequence, the proposed pKAS scheme can achieve offline password change.

6.2. Performance Analysis. In this section, we compare our scheme with similar schemes in terms of security performance, communication consumption, and computing consumption. The results indicate that pKAS is more secure and effective than other similar schemes. In addition, the presented pKAS has lower communication and computation costs.

6.2.1. Comparison of Security Features. We define $F_1$, $F_2$, $F_3$, $F_4$, $F_5$, $F_6$, $F_7$, $F_8$, $F_9$, $F_{10}$, $F_{11}$, and $F_{12}$ are the functionality of “be secure against offline password guessing attack,” “be secure against online password guessing attack,” “provide anonymous interactions,” “provide forward secrecy,” “be secure against forgery attack,” “provide mutual authentication,” “be secure against replay attack,” “be secure against parallel attack,” “be secure against insider attack,” “achieve user untraceability,” “achieve key agreement,” and “achieve offline password change,” respectively. In Table 2, we compare the security features of pKAS with related scheme, such as Irshad et al. [33], Tsai and Lo [35], and Kaur et al. [36].

6.2.2. Comparison of the Computation Cost. It is more convenient to define $T_{BP}$, $T_{ME}$, $T_{PM}$, $T_{PA}$, and $T_{HO}$ are the running time (in ms) of a single bilinear pairing operation, modular exponentiation operation, elliptic curve point multiplication, point addition, and hash operation, respectively. In Table 3, we list the computing time of the server and the mobile terminal separately. The cost in Table 3 is based on [36]. We use simulation Alibaba’s cloud server, and its configuration is Intel(R) Xeon(R) CPU E5-2630@2.30 GHz, 1 GB RAM and Ubuntu 14.04. In addition, the smartphone we use is configured with 2 GHz ARM CPU armeabi-v7a, 300 MiB RAM and Android 4.4 to simulate the mobile terminal.
According to the time computation by each operation in Table 3, we compared the time in [33, 35, 36, and pKAS schemes, as shown in Table 4.

6.2.3. Comparison of the Communication Cost. The comparison results in Table 4 are based on assumptions such as result of hash function to be 160 bits, random number to be 128 bits, identifier to be 64 bits, time stamp to be 32 bits, and encryption/decryption and ECC point to be 320 bits. Table 5 shows a comparison of the communication cost between pKAS and other schemes [33, 35].

In summary, the presented pKAS which consumes lower communication and calculations than [33, 35]. Though the cost of [36] is lower than pKAS, the scheme cannot be secure against forgery attacks and insider attack, and its bandwidth consumption is relatively large. Furthermore, pKAS is more secure than [33, 35, 36]. So, pKAS is more suitable for user and server to verify each other.

7. Conclusion and Future Work

Aiming at the practical problems encountered in the key agreement between the user and server in the edge cloud computing environment, we propose a new password-based key agreement scheme. We use ECDLP to construct user anonymity and forward secrecy. By comparing security, communication, and calculation costs, the proposed pKAS has better security and lower cost. Furthermore, pKAS also meets all 12 security requirements.

Although pKAS is more secure and efficient than similar schemes, the lightweight key agreement scheme, such as no point multiply operation, is more favored. It is very challenging to design a secure and lightweight scheme. This will be the direction of our next research.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Conflicts of Interest

P. Liu is currently a lecturer at the Department of Computer Technology and Application, Qinghai University, Xining. Her research interest includes network protocol and protocol security (e-mail: 247750940@qq.com). Syed Hamad Shirazi is currently an Assistant Professor at the Department of Information Technology, Hazara University, Baffa, Pakistan. His research interest includes image processing and image security (syedhamad@hu.edu.pk). W. Liu is currently an assistant at the Department of Computer Technology and Application, Qinghai University, Xining. Her research interest includes network protocol and protocol security (e-mail: 1007759705@qq.com). Y. Xie is currently a Professor at the Department of Computer Technology and Application, Qinghai University, Xining. His research interest includes network protocol and protocol security (e-mail: mark.y.xie@qq.com).
Acknowledgments

This study was supported in part by the Science and Technology Foundation of Qinghai under grant no. 2019-ZJ-7065, the National Natural Science Foundation of China under grant no. 61572570, and Course Construction of Qinghai University under grant no. SZ19014.

References


